

## 明 細 書

## 画像処理装置および画像処理方法

## 技術分野

[0001] 本発明は、グラフィック演算処理装置等における画像処理技術に関する。

## 背景技術

[0002] 近年のコンピュータゲームやデジタル放送などの分野に利用されるコンピュータグラフィックス技術や画像処理技術の著しい進歩に伴い、3次元グラフィックなどをより精緻に画面表示することが可能となっている。3次元グラフィック処理においては、3次元の座標データを有する3次元オブジェクトを2次元のスクリーンに投影し、ディスプレイなどに表示する処理が行われる。

[0003] 3次元オブジェクトは、多角形、たとえば三角形のポリゴンの組み合わせによってモデリングされるのが通常である。この3次元オブジェクトを、2次元のスクリーンに投影する際には、各ポリゴンの頂点データからポリゴン内部に含まれるピクセルの輝度などの値を算出するラスタライズ処理が行われる。

このラスタライズ処理においては、DDA (Digital Differential Analyzer) と呼ばれる線形補間手法が用いられる。DDAによれば、頂点データからポリゴンの辺方向へのデータの傾きを求め、この傾きを用いて辺上のデータを算出する。続いてラスタライズ方向の傾きを算出して内部のピクセルを発生させていく。

[0004] ここで、特許文献1には、ピクセルを発生させる際に、所定の矩形領域に含まれる複数のピクセルからなるピクセル群を1つの処理単位とし、このピクセル群をまとめて後段の処理ブロックに送出することで、描画速度の向上を図る技術が開示されている。

[0005] 特許文献2においては、特許文献1に記載の技術を改良し、複数のピクセル群をひとつのピクセル群に統合して後段の処理ブロックに送出することにより、処理の効率化を図っている。

[0006] 特許文献1:特開2000-338959号公報

特許文献2:特開2003-123082号公報

## 発明の開示

### 発明が解決しようとする課題

- [0007] 近年、3次元コンピュータグラフィックの演算処理を行う画像処理装置の高性能化にともない、3次元オブジェクトをより詳細に描画するために、3次元オブジェクトを構成するポリゴンのサイズは小さくなる傾向にある。したがって、有効なピクセル数をわずかにしか含まないピクセル群が生成される可能性が高くなるため、より効率的な矩形領域すなわちピクセル群の統合処理が望まれる。
- [0008] 本発明はこのような課題に鑑みてなされたものであり、その目的は、より効率的に描画処理を実現する画像処理装置の提供にある。

### 課題を解決するための手段

- [0009] 本発明のある態様は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理技術に関する。この画像処理技術では、単位図形をスクリーン座標上で複数の単位領域に分割して出力する。分割して出力される単位領域を複数のサブ領域に分割する。さらに、分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する。破棄処理の結果残ったサブ領域に対して描画処理を行う。

### 発明の効果

- [0010] 本発明に係る画像処理技術によれば、単位図形の描画処理を効率的に行うことができる。

### 図面の簡単な説明

- [0011] [図1]第1の実施の形態に係る画像処理装置の構成を示す図である。
- [図2]ラスターライズ部におけるピクセル生成の様子を示す図である。
- [図3]図3(a)～(c)は、複数のスタンプが異なる条件下で統合処理される様子を示す図である。
- [図4]スタンプがクアッドに分割される様子を示す図である。
- [図5]第1の実施の形態に係る統合処理部におけるスタンプの統合処理のフローチャートである。

[図6]図5に示したフローチャートに従ってスタンプ同士が統合される様子を示す図である。

[図7]領域書込部の構成を示す図である。

[図8]領域書込部において、クアドが各出力ユニットに分配される様子を示す図である。

[図9]図9(a)～(c)は、領域書込部がキャッシュメモリに統合スタンプを書き込む様子を示す図である。

[図10]スタンプ統合処理の別の例を示す図である。

[図11]領域書込部において、クアドが各出力ユニットに分配される様子を示す図である。

[図12]図12(a)～(c)は、領域書込部が、図10の統合スタンプをキャッシュメモリに書き込む様子を示す図である。

[図13]第2の実施の形態において、異なるスタンプアドレスを有するスタンプが連続して入力された場合の統合処理を示す図である。

[図14]領域書込部において、クアドが各出力ユニットに分配される様子を示す図である。

[図15]図15(a)～(c)は、領域書込部が、図13の統合スタンプをキャッシュメモリに書き込む様子を示す図である。

#### 符号の説明

- [0012] 1000 画像処理装置、100 ラスタライザ、200 グラフィックプロセッサ、300 メインプロセッサ、400 メインメモリ、10 ラスタライズ部、20 領域分割部、30 領域破棄部、40 領域書込部、600 統合処理部、50 キャッシュメモリ、110 メモリインターフェース部、120 グラフィックメモリ、130 ディスプレイコントローラ。

#### 発明を実施するための最良の形態

- [0013] 実施の形態の詳細について説明する前に、その概要を述べる。

本発明のある態様は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理装置に関する。この画

[図6]図5に示したフローチャートに従ってスタンプ同士が統合される様子を示す図である。

[図7]領域書込部の構成を示す図である。

[図8]領域書込部において、クアドが各出力ユニットに分配される様子を示す図である。

[図9]図9(a)～(c)は、領域書込部がキャッシュメモリに統合スタンプを書き込む様子を示す図である。

[図10]スタンプ統合処理の別の例を示す図である。

[図11]領域書込部において、クアドが各出力ユニットに分配される様子を示す図である。

[図12]図12(a)～(c)は、領域書込部が、図10の統合スタンプをキャッシュメモリに書き込む様子を示す図である。

[図13]第2の実施の形態において、異なるスタンプアドレスを有するスタンプが連続して入力された場合の統合処理を示す図である。

[図14]領域書込部において、クアドが各出力ユニットに分配される様子を示す図である。

[図15]図15(a)～(c)は、領域書込部が、図13の統合スタンプをキャッシュメモリに書き込む様子を示す図である。

#### 符号の説明

- [0012] 1000 画像処理装置、100 ラスタライザ、200 グラフィックプロセッサ、300 メインプロセッサ、400 メインメモリ、10 ラスタライズ部、20 領域分割部、30 領域破棄部、40 領域書込部、600 統合処理部、50 キャッシュメモリ、110 メモリインターフェース部、120 グラフィックメモリ、130 ディスプレイコントローラ。

#### 発明を実施するための最良の形態

- [0013] 実施の形態の詳細について説明する前に、その概要を述べる。

本発明のある態様は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理装置に関する。この画

像処理装置は、単位図形をスクリーン座標上で複数の単位領域に分割して出力するラスタライズ部と、ラスタライズ部から出力される単位領域を複数のサブ領域に分割する領域分割部と、領域分割部において分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄部と、領域破棄部による破棄処理の結果残ったサブ領域をメモリに書き込む領域書込部と、を備える。

[0014] この態様によれば、単位領域をサブ領域に分割し、不要なサブ領域を破棄することにより、後段に送出されるサブ領域数を減少させることができ、描画処理を効率的に行うことができる。単位図形とは、ポリゴンや曲面パッチをいう。

[0015] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域を再統合し、再統合により得られる統合領域ごとに、サブ領域をメモリに書き込んでよい。

[0016] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域のうち、分割前にサブ領域が属していた単位領域のスクリーン座標系における座標が互いに同一のサブ領域を統合してもよい。

この場合、再統合後の統合領域に含まれるサブ領域は、すべて分割前に属していた単位領域が同一位置となる。その結果、メモリに当該統合領域に関するデータを書き込む際に、各サブ領域ごとに、単位領域の座標を参照する必要がなくなる。

[0017] 統合領域は、単位領域と同サイズを有してもよい。この場合、ラスタライズ部の出力データと領域書込部の出力データのサイズが同一となる。ラスタライズ部における単位処理量と、メモリに書き込む際の単位処理量とを等しくすることにより、ハードウェア、ソフトウェア設計を柔軟に行うことができ、また既存のシステムとの整合性を高めることができる。

[0018] サブ領域のサイズは、領域書込部が当該サブ領域をメモリに書き込む際の単位処理量に相当してもよい。

[0019] 領域書込部は、書込対象となるサブ領域を、分割前にサブ領域が属していた単位領域内における相対的な位置情報を参照してメモリの該当アドレスに書き込んでよい。

[0020] 単位領域は矩形領域であって、ラスタライズ部は、複数の単位領域がそれぞれ、縦横のピクセル数が互いに等しいピクセル群を含むよう描画領域を分割し、領域分割

部は、ピクセル群を含む単位領域を、縦横のピクセル数が互いに等しい小ピクセル群を含む複数のサブ領域に分割してもよい。

- [0021] 領域破棄部は、領域分割部において分割された複数のサブ領域のうち、有効なピクセルを含まないサブ領域を対象として破棄処理を行ってもよい。「有効なピクセル」とは、単位図形に囲まれた領域に対応するピクセルであり、ラスタライズ部によって輝度値やフォグ値などが生成されたピクセルをいう。

有効なピクセルを含まないサブ領域を破棄することにより、後段の処理ユニットにおいて、無効なピクセルを処理する頻度が減少し、効率的な描画処理を行うことができる。

- [0022] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域のうち、スクリーン座標系において互いに同一の座標に有効なピクセルを含まないサブ領域を再統合し、再統合により得られる統合領域ごとに、サブ領域をメモリに書き込んでもよい。

スクリーン座標系において同一の座標に有効なピクセルを有するサブ領域同士を、再統合の対象から除外することにより、有効なピクセルが重複することによる情報の欠損を確実に防止することができる。

- [0023] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域のうち、分割前にサブ領域が属していた単位領域のスクリーン座標系における座標が互いに同一のサブ領域を統合してもよい。

この場合、再統合後の統合領域に含まれるサブ領域は、すべて分割前に属していた単位領域が同一位置となる。その結果、メモリに当該統合領域に関するデータを書き込む際に、各サブ領域ごとに、単位領域の座標位置を参照する必要がなくなる。

- [0024] 領域書込部は、書込対象となるサブ領域を、分割前にサブ領域が属していた単位領域内における相対的な位置情報を参照してメモリの該当アドレスに書き込んでもよい。

再統合によってサブ領域の相対的な位置情報が変化した場合においても、メモリに書き込む際に、もとの単位領域における相対的な位置情報を参照することにより、適切なアドレスに書き込むことができる。

- [0025] 領域書込部は、サブ領域に含まれるピクセルを、並列的にメモリに書込むメモリアク

像処理装置は、単位図形をスクリーン座標上で複数の単位領域に分割して出力するラスタライズ部と、ラスタライズ部から出力される単位領域を複数のサブ領域に分割する領域分割部と、領域分割部において分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄部と、領域破棄部による破棄処理の結果残ったサブ領域をメモリに書き込む領域書込部と、を備える。

[0014] この態様によれば、単位領域をサブ領域に分割し、不要なサブ領域を破棄することにより、後段に送出されるサブ領域数を減少させることができ、描画処理を効率的に行うことができる。単位図形とは、ポリゴンや曲面パッチをいう。

[0015] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域を再統合し、再統合により得られる統合領域ごとに、サブ領域をメモリに書き込んでよい。

[0016] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域のうち、分割前にサブ領域が属していた単位領域のスクリーン座標系における座標が互いに同一のサブ領域を統合してもよい。

この場合、再統合後の統合領域に含まれるサブ領域は、すべて分割前に属していた単位領域が同一位置となる。その結果、メモリに当該統合領域に関するデータを書き込む際に、各サブ領域ごとに、単位領域の座標を参照する必要がなくなる。

[0017] 統合領域は、単位領域と同サイズを有してもよい。この場合、ラスタライズ部の出力データと領域書込部の出力データのサイズが同一となる。ラスタライズ部における単位処理量と、メモリに書き込む際の単位処理量とを等しくすることにより、ハードウェア、ソフトウェア設計を柔軟に行うことができ、また既存のシステムとの整合性を高めることができる。

[0018] サブ領域のサイズは、領域書込部が当該サブ領域をメモリに書き込む際の単位処理量に相当してもよい。

[0019] 領域書込部は、書込対象となるサブ領域を、分割前にサブ領域が属していた単位領域内における相対的な位置情報を参照してメモリの該当アドレスに書き込んでよい。

[0020] 単位領域は矩形領域であって、ラスタライズ部は、複数の単位領域がそれぞれ、縦横のピクセル数が互いに等しいピクセル群を含むよう描画領域を分割し、領域分割

部は、ピクセル群を含む単位領域を、縦横のピクセル数が互いに等しい小ピクセル群を含む複数のサブ領域に分割してもよい。

- [0021] 領域破棄部は、領域分割部において分割された複数のサブ領域のうち、有効なピクセルを含まないサブ領域を対象として破棄処理を行ってもよい。「有効なピクセル」とは、単位図形に囲まれた領域に対応するピクセルであり、ラスタライズ部によって輝度値やフォグ値などが生成されたピクセルをいう。

有効なピクセルを含まないサブ領域を破棄することにより、後段の処理ユニットにおいて、無効なピクセルを処理する頻度が減少し、効率的な描画処理を行うことができる。

- [0022] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域のうち、スクリーン座標系において互いに同一の座標に有効なピクセルを含まないサブ領域を再統合し、再統合により得られる統合領域ごとに、サブ領域をメモリに書き込んでもよい。

スクリーン座標系において同一の座標に有効なピクセルを有するサブ領域同士を、再統合の対象から除外することにより、有効なピクセルが重複することによる情報の欠損を確実に防止することができる。

- [0023] 領域書込部は、領域破棄部による破棄処理の結果残ったサブ領域のうち、分割前にサブ領域が属していた単位領域のスクリーン座標系における座標が互いに同一のサブ領域を統合してもよい。

この場合、再統合後の統合領域に含まれるサブ領域は、すべて分割前に属していた単位領域が同一位置となる。その結果、メモリに当該統合領域に関するデータを書き込む際に、各サブ領域ごとに、単位領域の座標位置を参照する必要がなくなる。

- [0024] 領域書込部は、書込対象となるサブ領域を、分割前にサブ領域が属していた単位領域内における相対的な位置情報を参照してメモリの該当アドレスに書き込んでもよい。

再統合によってサブ領域の相対的な位置情報が変化した場合においても、メモリに書き込む際に、もとの単位領域における相対的な位置情報を参照することにより、適切なアドレスに書き込むことができる。

- [0025] 領域書込部は、サブ領域に含まれるピクセルを、並列的にメモリに書込むメモリアク



セス部を備えてもよい。

- [0026] 本発明の別の態様は、画像処理方法である。この画像処理方法は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理方法であって、単位図形をスクリーン座標上で複数の単位領域に分割して出力するラスタライズステップと、ラスタライズステップにおいて出力される単位領域を複数のサブ領域に分割する分割ステップと、領域分割部において分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄ステップと、領域破棄部による破棄処理の結果残ったサブ領域をメモリに書き込む領域書込ステップと、を備える。
- [0027] ラスタライズステップにおいて、複数の単位領域がそれぞれ、縦横のピクセル数が互いに等しいピクセル群を含むよう描画領域を分割し、領域分割ステップにおいて、ピクセル群を含む単位領域を、縦横のピクセル数が互いに等しい小ピクセル群を含む複数のサブ領域に分割してもよい。
- [0028] 領域破棄ステップにおいて、分割された複数のサブ領域のうち、有効なピクセルを含まないサブ領域を対象として破棄処理を行ってもよい。
- [0029] 本発明の別の態様も描画処理方法である。この描画処理方法は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理方法であって、単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を抽出して統合領域を生成するステップと、を備える。
- [0030] 本発明の別の態様の描画処理方法は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理方法であって、単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を並行的にメモリに書き込むステップと、を備える。
- [0031] 本発明の別の態様はコンピュータプログラムに関する。このコンピュータプログラムは、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリー

ン座標系にて順次描画処理するコンピュータプログラムであって、コンピュータに、単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を抽出して統合領域を生成するステップと、を実行させる。

[0032] 本発明の別の態様のコンピュータプログラムは、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理するコンピュータプログラムであって、コンピュータに、単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を並行的にメモリに書き込むステップと、を実行させる。

[0033] 本発明の別の態様は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理装置である。この画像処理装置は、単位図形をスクリーン座標に投影した状態で、スクリーンに対応する描画領域を複数の単位領域に分割して出力し、第2の単位図形以降についても同様の処理を経て、各単位図形を構成する複数の単位領域を単位図形ごとに順次出力するラスタライズ部と、ラスタライズ部から順次出力される単位領域を複数のサブ領域に分割する領域分割部と、前記領域分割部において分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄部と、領域破棄部による破棄処理の結果残ったサブ領域をメモリに書き込む領域書込部と、を備える。

[0034] 本発明の別の態様は、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理方法である。この画像処理方法は、第1の単位図形をスクリーン座標に投影した状態で、スクリーンに対応する描画領域を複数の合同な単位領域に分割して生成し、第2の単位図形以降についても同様の処理を経て、各単位図形を構成する複数の単位領域を単位図形ごとに順次出力するラスタライズステップと、ラスタライズステップにおいて、順次生成される前記単位領域を複数のサブ領域に分割する領域分割ステップと、領域分割ステップにおいて分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄ステップと、領域破棄ステップによる破棄処理の結果残ったサブ領

域をメモリに書き込む領域書込ステップと、を備える。

[0035] 本発明のさらに別の態様は、コンピュータプログラムに関する。このコンピュータプログラムは、描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理装置において、コンピュータに、第1の単位図形をスクリーン座標に投影した状態で、スクリーンに対応する描画領域を複数の単位領域に分割して生成し、第2の単位図形以降についても同様の処理を経て、各単位図形を構成する複数の単位領域を単位図形ごとに順次生成するラスタライズステップと、ラスタライズステップにおいて順次生成される単位領域を複数のサブ領域に分割する領域分割ステップと、領域分割ステップにおいて分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄ステップと、破棄処理の結果残ったサブ領域をメモリに書き込む領域書込ステップと、を実行させる。

[0036] なお、以上の構成要素の任意の組合せや本発明の構成要素や表現を方法、装置、システム、プログラムなどの間で相互に置換したものもまた、本発明の態様として有効である。

[0037] 以下、これらの発明の態様について、実施の形態をもとに詳細に説明する。

[0038] (第1の実施の形態)

図1は、第1の実施の形態に係る画像処理装置の構成を示す。画像処理装置1000は、3次元コンピュータグラフィックなどの画像処理を行い、描画対象のオブジェクトのサーフェスを構成する単位図形をスクリーン座標系の描画領域に投影し、ピクセルを発生させて描画処理を行い、ディスプレイに表示する。

[0039] 画像処理装置1000は、グラフィックプロセッサ200、メインプロセッサ300、メインメモリ400、グラフィックメモリ120を含む。これらのブロックは、バス500を介して接続されている。

画像処理装置1000は、図示しない表示装置と接続されており、画像処理装置1000により生成された画像、映像を出力する。図1を含む以降の図において、様々な処理を行う機能ブロックとして記載される各要素は、ハードウェア的には、CPU、メモリ、その他のLSIで構成することができ、ソフトウェア的には、メモリにロードされた予約管理機能のあるプログラムなどによって実現される。したがって、これらの機能ブロックが

ハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは当業者には理解されるところであり、いずれかに限定されるものではない。

[0040] メインプロセッサ300は、3次元コンピュータグラフィックスのモデリングなどの演算処理を行う。メインメモリ400は、主にメインプロセッサ300によって使用される記憶領域である。このメインメモリ400は、例えば、メインプロセッサ300によりコンピュータグラフィックスに関するタスクを処理して得られたモデリングデータなどが一時的に格納される。

[0041] グラフィックメモリ120は、グラフィックプロセッサ200により使用、管理されるグラフィックに関するデータ専用のメモリ領域である。このグラフィックメモリ120は、画像フレームデータが格納されるフレームバッファやZバッファに加え、画像フレームデータを描画する際に参照される基本データである頂点データ、テクスチャデータ、カラー.Lookupアップテーブルなどが格納される領域を含んでいる。

[0042] グラフィックプロセッサ200は、画像に関する処理を専用に行うブロックであり、メインプロセッサ300により生成された3次元モデリングデータをメインメモリ400から読み出し、3次元モデリングデータをもとに座標変換、陰面消去、シェーディングを行って画像フレームデータを作成する一連のレンダリング処理などを行う。

グラフィックプロセッサ200は、ラスタライザ100、メモリインターフェース部110、ディスプレイコントローラ130を含む。

[0043] ラスタライザ100は、メインメモリ400から3次元モデリングデータを読み出し、描画対象となるプリミティブの頂点データを取得する。ラスタライザ100は、この頂点データをもとに、3次元空間上のプリミティブを、スクリーン座標系への投影により描画平面上の図形に変換するビュー変換を行う。さらに、描画平面上の図形を、描画平面の水平方向に沿ってスキャンしながら、一列ごとに量子化されたピクセルに変換するラスタ処理を行う。このラスタライザ100によってプリミティブがピクセル展開されて、各ピクセル毎にピクセル情報が算出される。このピクセル情報には、RGBカラー値、透明度を表す $\alpha$ 値、視点からの奥行きを表すZ値が含まれる。

ラスタライザ100は、スキャンラインに沿って所定の大きさのピクセル領域を生成し、

メモリインターフェース部110へと出力する。ラスタライザ100から出力されるピクセル群は、一度キューにスタックされ、メモリインターフェース部110はスタックされたピクセル群を順にグラフィックメモリ120に書き込んでいく。

[0044] メモリインターフェース部110は、グラフィックメモリ120にピクセルを書込み、あるいはディスプレイコントローラ130からの指示によってグラフィックメモリ120からフレームバッファを読み出す。ディスプレイコントローラ130は、グラフィックメモリ120から読み出された画像データを表示装置へと出力する。

[0045] 以下、ラスタライザ100の構成および動作について詳細に説明する。

ラスタライザ100は、ラスタライズ部10、統合処理部600、キャッシュメモリ50を含む。

[0046] ラスタライズ部10は、描画対象のオブジェクトのサーフェスを構成する単位図形である三角形をスクリーン座標系の描画領域に投影し、描画領域を複数の単位領域に分割し、当該単位領域に含まれるピクセルを生成して、単位領域ごとにピクセル群として出力する。ラスタライズ部10は、複数の単位図形(三角形)ごとに、順次単位領域をピクセル群として生成し、出力する。

[0047] 図2は、ラスタライズ部10におけるピクセル生成の様子を示す図である。図2は、スクリーン座標系の描画領域2000の一部を示している。三角形12は、3次元オブジェクトのサーフェスを構成する単位図形である1つの三角形が描画領域に投影されたものである。ラスタライズ部10は、この三角形12の3頂点12a、12b、12cの座標値、色、フォグ値、テクスチャ座標などにもとづいて、ポリゴン内部のピクセルを、DDA(Digital Differential Analyze)法によって生成する。

[0048] 描画領域は、合同な複数の単位領域14に分割される。

ラスタライズ部10は、各単位領域ごとに順次、三角形12に囲まれる領域のピクセル情報(色、フォグ値など)を生成し、単位領域ごとに生成したピクセル情報を出力する。以下、ラスタライズ部10において、ピクセル情報の生成がなされたピクセルを、有効なピクセルという。

[0049] 本実施の形態において、所定のサイズを有する各単位領域14は、縦横4ピクセル×4ピクセルを含んでいる。この単位領域14に含まれる縦横4×4のピクセル群をスタ

メモリインターフェース部110へと出力する。ラスタライザ100から出力されるピクセル群は、一度キューにスタックされ、メモリインターフェース部110はスタックされたピクセル群を順にグラフィックメモリ120に書き込んでいく。

[0044] メモリインターフェース部110は、グラフィックメモリ120にピクセルを書込み、あるいはディスプレイコントローラ130からの指示によってグラフィックメモリ120からフレームバッファを読み出す。ディスプレイコントローラ130は、グラフィックメモリ120から読み出された画像データを表示装置へと出力する。

[0045] 以下、ラスタライザ100の構成および動作について詳細に説明する。

ラスタライザ100は、ラスタライズ部10、統合処理部600、キャッシュメモリ50を含む。

[0046] ラスタライズ部10は、描画対象のオブジェクトのサーフェスを構成する単位図形である三角形をスクリーン座標系の描画領域に投影し、描画領域を複数の単位領域に分割し、当該単位領域に含まれるピクセルを生成して、単位領域ごとにピクセル群として出力する。ラスタライズ部10は、複数の単位図形(三角形)ごとに、順次単位領域をピクセル群として生成し、出力する。

[0047] 図2は、ラスタライズ部10におけるピクセル生成の様子を示す図である。図2は、スクリーン座標系の描画領域2000の一部を示している。三角形12は、3次元オブジェクトのサーフェスを構成する単位図形である1つの三角形が描画領域に投影されたものである。ラスタライズ部10は、この三角形12の3頂点12a、12b、12cの座標値、色、フォグ値、テクスチャ座標などにもとづいて、ポリゴン内部のピクセルを、DDA (Digital Differential Analyze) 法によって生成する。

[0048] 描画領域は、合同な複数の単位領域14に分割される。

ラスタライズ部10は、各単位領域ごとに順次、三角形12に囲まれる領域のピクセル情報(色、フォグ値など)を生成し、単位領域ごとに生成したピクセル情報を出力する。以下、ラスタライズ部10において、ピクセル情報の生成がなされたピクセルを、有効なピクセルという。

[0049] 本実施の形態において、所定のサイズを有する各単位領域14は、縦横4ピクセル×4ピクセルを含んでいる。この単位領域14に含まれる縦横4×4のピクセル群をスタ

ンプSTPと呼ぶ。各スタンプSTPは、対応する単位領域のスクリーン座標系における位置情報を保持する。以下、スタンプの位置情報をスタンプアドレスADDstpという。

ラスタライズ部10は、スタンプSTP単位で順次ピクセルの生成を行い、有効なピクセルを含むスタンプSTPを順次統合処理部600へと出力する。

[0050] 統合処理部600は、ラスタライズ部10から出力される複数のスタンプSTPを統合し、統合されたスタンプごとに後段のキャッシュメモリ50に書き込む。統合処理部600において複数のスタンプSTPを統合して得られるスタンプを最終スタンプSTPfという。グラフィックプロセッサ200内部において、ラスタライザ100およびメモリインターフェース部110は、スタンプSTPを1つの処理単位として各ブロック間でデータの送受を行っている。

[0051] はじめに、統合処理部600における統合処理について説明する。統合処理部600は、互いにスタンプアドレスADDstpが同一のスタンプ同士を統合処理する。

スタンプ同士の統合処理は、以下の2つをパラメータとして行われる。第1パラメータは、統合可能かを判定する単位の大きさ、第2パラメータは、実際に統合処理を行う単位の大きさである。図3(a)～(c)は、複数のスタンプが異なる条件下で統合処理される様子を示す図である。スタンプSTPmは、2つのスタンプSTP1、STP2を統合して得られる統合スタンプSTPmを表している。

以下、本実施の形態において、スタンプを分割して得られる縦横2×2ピクセルの小ピクセル群をクアッドQDと呼ぶ。また、有効なピクセルを含むクアッドを有効なクアッドと呼ぶ。

[0052] 図3(a)は、2つのパラメータをいずれもピクセルとした場合の統合処理を示す。図3(a)のスタンプSTP1およびスタンプSTP2は、いずれも同一のスタンプアドレスを有している。これらのスタンプSTPは、互いに重複した位置に有効なピクセルを含まない。したがって、ピクセルごとに重複が判定された結果、問題なく統合される。

[0053] 図3(b)は、2つのパラメータをいずれもクアッドQDとした場合の統合処理を示す。図3(b)の統合処理においては、クアッドを単位として重複が判定される。その結果、スタンプSTP1とスタンプSTP2の右上のクアッドはいずれも有効であり、重複しているため、2つのスタンプは統合されず、2つのスタンプSTP1、STP2がそのまま出力さ

れる。

- [0054] 図3(c)は、本実施の形態に係る統合処理であり、第1パラメータをピクセルとし、第2パラメータをクアッドQDとした場合の統合処理を表す。図3(c)の統合処理においては、ピクセルを単位として重複が判定される。その結果、スタンプSTP1とスタンプSTP2の右上のクアッドQDはいずれも有効で重複しているが、ピクセル単位の判定では、有効なピクセルが重複しておらず、かつ2つのスタンプに含まれる有効なクアッドの総数は4つ以下なので1つのスタンプSTPmに統合することができる。なお、クアッド単位で統合処理を行った結果得られる統合スタンプSTPmは、各クアッドの位置が元の位置から変更される場合があるため、後述の再配置を行う。

このように、図3(c)に示す本実施の形態に係る統合処理においては、重複の判定はピクセル単位で行い、統合処理はクアッド単位で行う。その結果、図3に示したスタンプSTP1とスタンプSTP2を統合する場合、統合処理の効率は、図3(a)の場合と同等となる。実際の画像処理においても、統合処理の効率は、図3(a)、図3(c)、図3(b)の順で優れている。

- [0055] また、このような統合処理を行うための画像処理装置において、その実装コストについて考察すると、第1パラメータ、すなわち統合可能かを判定する単位を小さく設定する場合、重複を判定するユニットのみを変更すればよいため、実装コストはそれほど上昇しない。一方、第2パラメータ、すなわち、実際に統合処理を行う単位を小さくすると、メモリにアクセスするユニットの実装コストが高くなる。したがって、実装コストの観点から見ると、図3(b)、図3(c)、図3(a)の順で優れているといえる。

- [0056] このように、図3(c)に示した本実施の形態に係る統合処理方法は、統合処理効率と実装コストのバランスに優れている。以下、本実施の形態に係る統合処理を実現するための統合処理部600の構成について説明する。

- [0057] 統合処理部600は、領域分割部20、領域破棄部30、領域書込部40を含む。

領域分割部20は、ラスライズ部10から順次出力されるスタンプSTPを、それぞれ縦横のピクセル数が等しい合同な複数の小ピクセル群に分割する。本実施の形態において、領域分割部20は、縦横4×4ピクセルのスタンプSTPを、縦横2×2ピクセルの合同なクアッドQD0～QD3に分割して、後段の領域破棄部30へと出力する。



[0058] 図4は、スタンプSTPがクアッドQD0～QD3に分割される様子を示す図である。各クアッドQD0～QD3は、スタンプ内における相対的な位置情報をクアッドアドレスADDqdとして保持する。クアッドアドレスADDqdは、左上のクアッドから右下のクアッドに向かってそれぞれ、00、01、10、11であるとする。図4において、クアッドアドレスは括弧内に示している。

領域分割部20は、複数のピクセルをクアッドQD単位で後段の領域破棄部30へと出力する。

[0059] 領域破棄部30は、領域分割部20から出力される複数のクアッドQDを所定の規則のもと間引いて破棄する。領域破棄部30は、破棄処理の結果残ったクアッドQDを後段の領域書込部40に出力する。

[0060] 領域書込部40は、領域破棄部30における破棄処理の結果残ったクアッドQDをキャッシュメモリ50に書き込む。このとき、領域破棄部30による破棄処理の結果残ったクアッドQDは再統合され、縦横4×4ピクセルで構成されるスタンプに再構成される。領域書込部40により再統合されたスタンプを統合スタンプSTPmという。

[0061] 領域書込部40は、領域破棄部30から出力された統合スタンプSTPmを処理単位としてキャッシュメモリ50に書き込む。統合スタンプSTPmは、キャッシュメモリ50に書き込まれることにより、キューにスタックされ、メモリインターフェース部110はスタックされた統合スタンプSTPmを順にグラフィックメモリ120に書き込んでいく。統合スタンプSTPmと最終スタンプSTPfは必ずしも同一とは限らないため、領域書込部40は、後述の結合スタンプSTPm内のクアッドの再配置を行う。

[0062] 次に、領域分割部20、領域破棄部30、領域書込部40における処理について詳細に説明する。

ラスライズ部10から順次出力されるスタンプSTPのうち、領域分割部20において分割処理の対象となるスタンプをソーススタンプSTPsという。ソーススタンプSTPsを順次、一旦クアッドQDに分割し、不要なクアッドQDを破棄した後に、再統合した結果得られるスタンプを統合スタンプSTPmという。また、ソーススタンプSTPsを統合する過程において使用され、統合処理を受けるべく保持されるスタンプをターゲットスタンプSTPtという。

統合処理部600においては、ターゲットスタンプSTP<sub>t</sub>の空クアッドに、ソーススタンプSTPsのクアッドQDsを順次書き込むことにより統合処理が進められる。

[0063] 図5は、本実施の形態に係る統合処理部600におけるスタンプSTPの統合処理のフローチャートである。

[0064] まず、統合処理の開始に先立ち、領域書込部40において、ターゲットスタンプSTP<sub>t</sub>を初期化する(S100)。

次に変数jを0に初期化する(S102)。変数jは、ターゲットスタンプSTP<sub>t</sub>の何番目のクアッドまで有効なクアッドが書き込まれているかを表す。たとえば、j=0のとき、ターゲットスタンプSTP<sub>t</sub>には、1つのクアッドも書き込まれていないことを表し、j=4となると、ターゲットスタンプSTP<sub>t</sub>のすべてのクアッドが埋まっていることを示す。

[0065] 次に領域分割部20は、ソーススタンプSTPsを取得する(S104)。ソーススタンプSTPsの取得は、ラスタライズ部10から出力されたスタンプSTPを読み込むことにより行われる。

[0066] 次にターゲットスタンプSTP<sub>t</sub>とソーススタンプSTPsのスタンプアドレスADD<sub>stp</sub>が比較される(S106)。その結果、2つのスタンプのアドレスが異なる場合(S106のN)には統合処理を行わず、現在のターゲットスタンプSTP<sub>t</sub>を統合スタンプSTP<sub>m</sub>として後段の領域書込部40へ出力し(S130)、ターゲットスタンプSTP<sub>t</sub>を初期化する(S132)。ターゲットスタンプSTP<sub>t</sub>の初期化に伴い、変数jを0とする(S134)。その後、S108に戻る。

[0067] 一方、2つのスタンプのアドレスが同一の場合(S106のY)には、領域分割部20は、ソーススタンプSTPsを4つのクアッドQDs0～QDs3に分割する(S108)。

次いで変数iを0に初期化する(S110)。変数iは、統合対象となるソーススタンプSTPsの何番目のクアッドが処理対象かを表す。たとえば、i=0のとき、ソーススタンプSTPsのクアッドQDs0が処理対象となる。

[0068] 次に、領域破棄部30は、ソーススタンプSTPsのi番目のクアッドQD<sub>si</sub>が有効なピクセルを含むかを判定する(S112)。その結果、クアッドQD<sub>si</sub>が有効なピクセルを含まない場合(S112のN)、そのクアッドQD<sub>si</sub>を破棄する(S160)。その後、次のクアッドQD<sub>si</sub>について処理すべく変数iを1増加する(S126)。

- [0069] 領域書込部40は、クアッドQDsiが有効なピクセルを含む場合(S112のY)、そのクアッドQDsiの有効なピクセルがターゲットスタンプSTPtの有効なピクセルと重複するかを判定する(S114)。この判定は、クアッドQDsiとターゲットスタンプSTPtが、スクリーン座標系において互いに同一の座標位置に有効なピクセルを含むかを判定することにより行われる。
- [0070] その結果、互いに有効なピクセルが重複する場合(S114のY)には、統合処理は行わず、それまでのターゲットスタンプSTPtを統合スタンプSTPmとして出力し、ターゲットスタンプSTPtを初期化し、さらに変数jを0に戻す(S140～S144)。その後、以下の統合処理(S116～S128)に進む。
- また、有効なピクセルが重複しない場合(S114のN)にも、以下の統合処理(S116～S128)に進む。
- [0071] ソーススタンプSTPsのクアッドQDsiを、ターゲットスタンプSTPtのj番目のクアッドQDtjに書き込む(S116)。その後、次のターゲットスタンプSTPtの書き込み位置をずらすために、変数jを1増加する(S118)。
- ここでj=4ならば(S120のY)、ターゲットスタンプSTPtのすべてのクアッドQDt0～QDt3にソーススタンプSTPsのクアッドQDsが統合されたことになるため、現在のターゲットスタンプSTPtを統合スタンプSTPmとして出力する(S150)。次いで、ターゲットスタンプSTPtを初期化し(S152)、変数jを0とする(S154)。次のクアッドQDsiの統合処理を行うため、変数iを1増加する(S126)。
- [0072] j≠4ならば(S120のN)、ターゲットスタンプSTPtに空クアッドが存在するため、ターゲットスタンプSTPtはそのままとし、次のクアッドQDsiの統合処理を行うために、変数iを1増加する(S126)。
- [0073] 次に、変数iについて判定を行い、i=4ならば(S128のY)、ソーススタンプSTPsに属するすべてのクアッドQDsiについて統合処理を終えたことになるため、次のソーススタンプSTPsを取得する(S104)。
- [0074] i≠4ならば(S128のN)、次のクアッドQDsiについて統合処理を行うべく、S112に戻る。
- [0075] このようにして、統合処理部600は、ターゲットスタンプSTPtとソーススタンプSTP

mのスタンプアドレスが同一で、かつ同一ピクセル位置に有効なピクセルが存在しない場合に統合する。

- [0076] 図6は、図5に示したフローチャートに従ってスタンプ同士が統合される様子を示す図である。いま、スタンプSTP1～STP4が順番に領域分割部20に入力される場合を考える。初期状態として、ターゲットスタンプSTPtが空スタンプであるとする。

まず、スタンプSTP1がソーススタンプSTPsとされ、領域分割部20によって4つのクアッドQDs0～QDs3に分割される。クアッドQDs1～QDs3は、有効なピクセルを含まないため、領域破棄部30により破棄される。

領域書込部40には、領域破棄部30における破棄処理の結果残ったクアッドQDs0のみ出力され、ターゲットスタンプSTPtのクアッドQDt0に書き込まれる。

- [0077] 次に、スタンプSTP2がソーススタンプSTPsとされ、領域分割部20によって4つのクアッドQDs0～QDs3に分割され、領域破棄部30によってクアッドQDs1～QDs3は破棄される。領域書込部40は、クアッドQDs0をターゲットスタンプSTPtのクアッドQDt1に書き込む。

スタンプSTP2、STP3についても、同様の処理が行われ、ターゲットスタンプSTPtのクアッドQDt0～QDt3には、それぞれスタンプSTP1～STP4のクアッドアドレスADDqdが(00)のクアッドQD0が書き込まれ、統合スタンプSTPmが生成される。

- [0078] 統合スタンプSTPmに含まれる各クアッドQDm0～QDm3は、もとのスタンプSTP1～STP4におけるクアッドアドレスを位置情報として保持している。統合スタンプSTPmは、この位置情報とともに、領域書込部40へと出力される。

- [0079] 図7は、領域書込部40の構成を示す図である。

領域書込部40は、メモリアクセス部44および分配部42を含む。メモリアクセス部44は、クアッドに含まれる4つのピクセルを、同時並列的にキャッシュメモリ50に書込む出力ユニット46a～46dを備える。

分配部42は、各出力ユニット46a～46dにクアッドQDを構成するピクセルを分配する。出力ユニット46aにはクアッドQD内の左上のピクセルが分配され、出力ユニット46bには右上、出力ユニット46cには左下、出力ユニット46dには右下のピクセルが分配される。

領域書込部40は、4つのピクセルを同時並列的にキャッシュメモリ50に書き込む動作を、クアッド単位で4回繰り返すことにより、1つの統合スタンプの書込処理を行う。

[0080] 図8は、領域書込部40において、クアッドQDm0～QDm3が各出力ユニット46a～46dに分配される様子を示す。分配部42は、統合スタンプSTPmを4つのクアッドQDm0～QDm3に分割し、時刻T1、T2、T3、T4と順番に、各出力ユニット46a～46dに分配する。

時刻T1において、分配部42によってクアッドQDm0がピクセルに分解され、4つの出力ユニット46a～46dに分配され、各出力ユニットは、分解されたピクセルを、それぞれキャッシュメモリ50に書き込む。次いで、時刻T2、T3、T4と各クアッドQDm1～QDm3がピクセルに展開されて順番に出力される。こうして、時刻T1～T4を1つの時間単位として、1つの統合スタンプSTPmが出力される。

[0081] 図9(a)～(c)は、領域書込部40がキャッシュメモリ50に統合スタンプSTPmを書き込む様子を示す図である。図9(a)は、キャッシュメモリ50を示し、図9(b)は、領域書込部40によるクアッドの書込状態を示し、図9(c)は、書込完了後の最終スタンプSTPfを示す。

[0082] 統合スタンプSTPmと最終スタンプSTPfは必ずしも同一とは限らないため、領域書込部40は、後述の結合スタンプSTPm内のクアッドの再配置を行う。

この再配置は以下のようにして行われる。

時刻T1に、領域書込部40は、統合スタンプSTPmのクアッドQDm0に含まれるピクセルをキャッシュメモリ50に書き込む。領域書込部40は、クアッドQDm0のクアッドアドレスADDqdを参照する。クアッドQDm0のクアッドアドレスADDqdは00であるため、キャッシュメモリ50のアドレス00に書き込まれる。

次に、時刻T2にクアッドQDm1を書き込む。クアッドQDm1のクアッドアドレスも00であるため、キャッシュメモリ50のアドレス00に書き込まれる。同様に、時刻T3、T4において、クアッドQDm2、QDm3がアドレス00に書き込まれる。

[0083] すべてのクアッドQDm0～QDm3がクアッドアドレスADDqd=00に書き込まれることにより、図6のスタンプSTP1～STP4が統合された最終スタンプSTPfが生成される。

- [0084] 図10は、別のスタンプSTP5～STP8が統合される様子を示す。これらのスタンプは、すべて同一のスタンプアドレスを有しており、また互いに重複する位置に有効なピクセルを有していない。
- [0085] まずスタンプSTP5が、4つのクアッドQDs0～QDs3に分割される。クアッドQDs0のみ有効なピクセルを含むため、ターゲットスタンプSTPtのクアッドQDt0に書き込まれ、クアッドQDs1～QDs3は破棄される。
- 次にスタンプSTP6をソーススタンプSTPsとし、クアッドQDs0～QDs3について統合処理が行われる。その結果、クアッドQDs0、QDs2が、ターゲットスタンプSTPtのクアッドQDt1、QDt2にそれぞれ書き込まれ、クアッドQDs1、QDs3が破棄される。
- [0086] 次にスタンプSTP7をソーススタンプSTPsとする。スタンプSTP7については、クアッドQDs0がターゲットスタンプSTPtのクアッドQDt3に書き込まれる。この状態で、ターゲットスタンプSTPt内のすべてのクアッドが有効となるため、統合スタンプSTPm1として出力し、ターゲットスタンプSTPtが初期化される。
- [0087] 続いて、ソーススタンプSTPsのクアッドQDs2が、新たなターゲットスタンプSTPtのクアッドQDt0に書き込まれる。クアッドQDs1、QDs3は破棄される。
- [0088] 次にスタンプSTP8をソーススタンプとする。クアッドQDs0～QDs3はすべて有効なピクセルを含むため、破棄されることなくターゲットスタンプSTPtに書き込まれる。
- クアッドQDs0～QDs2は、ターゲットスタンプSTPtのクアッドQDt1～QDt3にそれぞれ書き込まれる。この状態で、ターゲットスタンプSTPt内のすべてのクアッドが有効となるため、統合スタンプSTPm2として出力し、再度、ターゲットスタンプSTPtが初期化される。
- [0089] ソーススタンプSTPsのクアッドQDs3は、あらたなターゲットスタンプSTPtのクアッドQDt0に書き込まれる。その後、別のソーススタンプSTPsと統合され、統合スタンプSTPm3として出力される。
- [0090] 図11は、領域書込部40において、クアッドQDm0～QDm3が各出力ユニット46a～46dに分配される様子を示す。
- 領域書込部40は、時刻T1、T2、T3、T4と順番に、統合スタンプSTPm1内の4つのクアッドQDm0～QDm3を出力する。

領域書込部40において、分配部42は、各クアッドに含まれる4つのピクセルを、各出力ユニットに分配する。この統合スタンプSTPm1は、時刻T1～T4の期間に書き込まれ、統合スタンプSTPm2は時刻T5～T8、統合スタンプSTPm3は時刻T9～T12に書き込まれる。

各統合スタンプSTPm1～STPm3を構成する各クアッドQDm0～QDm3は、もとのスタンプSTP5～STP7内におけるアドレスとともに出力される。

[0091] 図12(a)～(c)は、領域書込部40が、図10に示す統合スタンプSTPm1～STPm3をキャッシュメモリ50に書き込む様子を示す図である。領域書込部40は、各時刻において、各クアッドをそのアドレスを参照して、再配置しながらキャッシュメモリ50に書き込まれていく。

その結果、キャッシュメモリ50には、図12(c)に示すように、図7のスタンプSTP5～STP8が統合された最終スタンプSTPfが書き込まれる。後段のメモリインターフェース部110は、この最終スタンプSTPfをグラフィックメモリ120に書き込む。

[0092] このように、本実施の形態に係る画像処理装置1000によれば、ラスタライズ部10から出力される処理単位となるピクセル群であるスタンプを、小ピクセル群であるクアッドに分割し、不要なクアッドを破棄し、残ったクアッド同士を統合する。その結果、統合後に得られる統合スタンプに含まれる有効なピクセル数を増加させることができ、描画処理の効率化を図ることができる。

[0093] 特に、トライアングルストリップの性質から、同一のスタンプアドレスを有し、かつ有効なピクセルが重ならないスタンプは連続して生成されやすい。したがって、これらの連続するスタンプを効率的に統合することにより、有効なピクセルを増加することができる。

また、本実施の形態に係る画像処理装置では、スタンプを一旦クアッドに分割し、ピクセル単位で統合可能かを判定しつつ、実際の統合処理はクアッド単位で行うため、実装コストと処理効率のバランスを良好に保つことができる。

[0094] さらに、領域破棄部30において統合された統合スタンプSTPmは、領域書込部40において、スタンプ内における局所的な配置の変更によって最終的に出力される最終統合スタンプSTPfに修正することができる。この配置の変更を行う領域書込部40

は、たとえば、キャッシュメモリ50のアドレスの制御、あるいはクロスバーによる配置変更などに相当するため、ハードウェア的な追加、修正はメモリアクセス部のみとなるため、従来の画像処理装置に対しても、容易に実装することができる。

[0095] (第2の実施の形態)

第2の実施の形態では、第1の実施の形態で説明したスタンプ統合処理を拡張したものである。第1の実施の形態では、統合処理の対象となるクアッドは、ソーススタンプSTPsがターゲットスタンプSTPmと同一のスタンプアドレスを有する場合のみとしたが、本実施の形態では、異なるスタンプアドレスを有するスタンプ間での統合処理も許可することを特徴とする。以下、異なるスタンプアドレスを有するスタンプ同士の統合を許可する統合処理を拡張統合処理という。

[0096] 図13は、異なるスタンプアドレスを有するスタンプが連続して入力された場合の統合処理を示す図である。領域分割部20には、スタンプSTP10～STP13がラスターライズ部10から順番に入力される。スタンプSTP10、STP11のスタンプアドレスADDstpは0001であり、スタンプSTP12、STP13のスタンプアドレスADDstpは0010であるとする。スタンプSTP10～STP13の有効なピクセルは、重複した位置には存在しない。

[0097] 領域分割部20は、スタンプSTP10をソーススタンプSTPsとし、クアッドQDs0～QDs3に分割する。クアッドQDs0がターゲットスタンプSTPtのクアッドQDt0に書き込まれ、クアッドQDs1～QDs3が破棄される。

次にスタンプSTP11をソーススタンプSTPsとし、クアッドQDs0～QDs3に分割した後、クアッドQDs0がターゲットスタンプSTPtのクアッドQDt1に書き込まれる。

次にスタンプSTP12をソーススタンプとし、クアッドQDs0～QDs3に分割した後、クアッドQDs2がターゲットスタンプSTPtのクアッドQDt2に書き込まれる。

次にスタンプSTP13をソーススタンプとしてクアッドQDs0～QDs3に分割し、クアッドQDs2がターゲットスタンプSTPtのクアッドQDt3に書き込まれる。

ターゲットスタンプSTPtに書き込まれなかったクアッドはすべて破棄される。

[0098] この状態でターゲットスタンプSTPtは、すべてのクアッドが有効となるため、統合スタンプSTPmとして出力され、ターゲットスタンプSTPtが初期化される。



[0099] 統合スタンプSTP<sub>m</sub>に含まれる各クアッドQD<sub>m</sub>0～QD<sub>m</sub>3は、もとのスタンプSTP10～STP13におけるクアッドアドレスADD<sub>qd</sub>とともに、もとのスタンプSTP10～STP13のスタンプアドレスADD<sub>stp</sub>を保持している。

図14は、領域書込部40において、クアッドQD<sub>m</sub>0～QD<sub>m</sub>3が各出力ユニット46a～46dに分配される様子を示す。領域破棄部30のメモリアクセス部44は、時刻T1、T2、T3、T4と順番に、統合スタンプSTP<sub>m</sub>内の4つのクアッドQD<sub>m</sub>0～QD<sub>m</sub>3を出力する。時刻T1において、クアッドQD<sub>m</sub>0がピクセルに展開され、同時並列的に出力される。

次いで、時刻T2、T3、T4と各クアッドQD<sub>m</sub>1～QD<sub>m</sub>3がピクセルに展開されて順番に出力される。

[0100] 図15(a)～(c)は、領域書込部40がキャッシュメモリ50に図13の統合スタンプSTP<sub>m</sub>を書き込む様子を示す図である。

図15(a)に示すように、領域書込部40は、各クアッドQD<sub>m</sub>を書き込む際に、クアッドアドレスADD<sub>qd</sub>に加えてスタンプアドレスADD<sub>stp</sub>を参照する。たとえば、時刻T1に入力されたクアッドQD<sub>m</sub>0は、スタンプアドレスが0001、クアッドアドレスは00であるため、キャッシュメモリ50の当該アドレス箇所に書き込まれる。クアッドQD<sub>m</sub>1～QD<sub>m</sub>3についても同様である。

その結果、図12(c)に示すように、キャッシュメモリ50には、図10のスタンプSTP10～STP13が統合された最終スタンプSTP<sub>f</sub>1、STP<sub>f</sub>2が書き込まれる。最終スタンプSTP<sub>f</sub>1は、スタンプSTP10、STP11が統合されたものであり、最終スタンプSTP<sub>f</sub>2は、スタンプSTP12、STP13が統合されたものである。最終スタンプSTP<sub>f</sub>1、STP<sub>f</sub>2は、それぞれスタンプアドレスADD<sub>stp</sub>が0001および0010の位置に書き込まれる。

[0101] このように、本実施の形態に係る拡張統合処理を行うことによって、異なるスタンプアドレスを有するスタンプ同士を一時的に統合することにより、領域書込部40においてキャッシュメモリ50にアクセスする回数を減らすことができる。

この拡張統合処理によれば、スタンプアドレスが異なっても統合処理が行うことができるため、統合スタンプ内の有効なクアッドの数を増加させ、画像処理の効率を

さらに向上することができる。

[0102] 上記実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

[0103] 実施の形態においては、ピクセル群を縦横4×4のスタンプで構成し、小ピクセル群を縦横2×2のクアッドとしたがこれには限定されず、小ピクセル群を縦横1×4、あるいは縦横4×1のクアッドとしてもよい。

クアッドに含まれるピクセルの数は、領域書込部40において同時並列的にキャッシュメモリ50に書き込まれるピクセルの数と同数であること望ましいが、かならずしもこれには限定されず、4ピクセルを同時並列的にキャッシュメモリ50に書き込む場合において、クアッドに含まれるピクセルの数を2つとしてもよい。スタンプのサイズ、クアッドのサイズは、ハードウェアコスト、処理に要するコスト等を考えて適宜決定すればよい。

#### 産業上の利用可能性

[0104] 本発明は、グラフィック演算処理装置等における画像処理技術に利用することができる。

### 請求の範囲

- [1] 描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとに、スクリーン座標系にて順次描画処理する画像処理装置であって、
- 単位図形をスクリーン座標上で複数の単位領域に分割して出力するラスタライズ部と、
- 前記ラスタライズ部から出力される単位領域を複数のサブ領域に分割する領域分割部と、
- 前記領域分割部における分割の結果得られた複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄部と、
- 前記領域破棄部による破棄処理の結果残ったサブ領域をメモリに書き込む領域書込部と、
- を備えることを特徴とする画像処理装置。
- [2] 前記領域書込部は、前記領域破棄部による破棄処理の結果残ったサブ領域を再統合し、再統合により得られる統合領域ごとに、前記メモリに書き込むことを特徴とする請求項1に記載の画像処理装置。
- [3] 前記統合領域は、前記単位領域と同サイズを有することを特徴とする請求項2に記載の画像処理装置。
- [4] 前記サブ領域のサイズは、前記領域書込部が当該サブ領域をメモリに書き込む際の単位処理量に相当することを特徴とする請求項1に記載の画像処理装置。
- [5] 前記領域書込部は、前記領域破棄部による破棄処理の結果残ったサブ領域のうち、分割前にサブ領域が属していた単位領域のスクリーン座標系における座標が互いに同一のサブ領域を統合することを特徴とする請求項2に記載の画像処理装置。
- [6] 前記領域書込部は、書込対象となるサブ領域を、分割前にサブ領域が属していた単位領域内における相対的な位置情報を参照して前記メモリの該当アドレスに書き込むことを特徴とする請求項2に記載の画像処理装置。
- [7] 前記単位領域は矩形領域であって、前記ラスタライズ部は、前記複数の単位領域がそれぞれ、縦横のピクセル数が互いに等しいピクセル群を含むよう描画領域を分割し、

前記領域分割部は、前記ピクセル群を含む前記単位領域を、縦横のピクセル数が互いに等しい小ピクセル群を含む複数のサブ領域に分割することを特徴とする請求項1に記載の画像処理装置。

- [8] 前記領域破棄部は、前記領域分割部において分割された複数のサブ領域のうち、有効なピクセルを含まないサブ領域を対象として破棄処理を行うことを特徴とする請求項7に記載の画像処理装置。
- [9] 前記領域書込部は、前記領域破棄部による破棄処理の結果残ったサブ領域のうち、スクリーン座標系において互いに同一の座標に有効なピクセルを含まないサブ領域を再統合し、再統合により得られる統合領域ごとに、前記メモリに書き込むことを特徴とする請求項7または8に記載の画像処理装置。
- [10] 前記領域書込部は、前記領域破棄部による破棄処理の結果残ったサブ領域のうち、分割前にサブ領域が属していた単位領域のスクリーン座標系における座標が互いに同一のサブ領域を統合することを特徴とする請求項9に記載の画像処理装置。
- [11] 前記領域書込部は、書込対象となるサブ領域を、分割前にサブ領域が属していた単位領域内における相対的な位置情報を参照して前記メモリの該当アドレスに書き込むことを特徴とする請求項9に記載の画像処理装置。
- [12] 前記領域書込部は、  
前記サブ領域に含まれるピクセルを、並列的に前記メモリに書き込むメモリアクセス部を備えることを特徴とする請求項7に記載の画像処理装置。
- [13] 描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理装置であって、  
単位図形をスクリーン座標上で複数の単位領域に分割して出力するラスタライズステップと、  
前記ラスタライズステップにおいて生成される前記単位領域を複数のサブ領域に分割する領域分割ステップと、  
前記領域分割ステップにおいて分割された複数のサブ領域のうち一部のサブ領域を所定の規則で破棄する領域破棄ステップと、  
前記領域破棄ステップによる破棄処理の結果残ったサブ領域をメモリに書き込む

領域書込ステップと、

を備えることを特徴とする画像処理方法。

- [14] 前記ラスタイズステップにおいて、前記複数の単位領域がそれぞれ、縦横のピクセル数が互いに等しいピクセル群を含むよう描画領域を分割し、
- 前記領域分割ステップにおいて、前記ピクセル群を含む前記単位領域を、縦横のピクセル数が互いに等しい小ピクセル群を含む複数のサブ領域に分割することを特徴とする請求項13に記載の画像処理方法。
- [15] 前記領域破棄ステップにおいて、分割された複数のサブ領域のうち、有効なピクセルを含まないサブ領域を対象として破棄処理を行うことを特徴とする請求項14に記載の画像処理方法。
- [16] 描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理方法であって、
- 単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、
- 前記出力するステップにおいて出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を抽出して統合領域を生成するステップと、
- を備えることを特徴とする画像処理方法。
- [17] 描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理する画像処理方法であって、
- 単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、
- 前記分割して出力するステップにおいて出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を並行的にメモリに書き込むステップと、
- を備えることを特徴とする画像処理方法。
- [18] 描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理するコンピュータプログラムであって、コンピュータに、
- 単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、
- 前記出力するステップにおいて出力される単位領域のそれぞれを構成する複数の

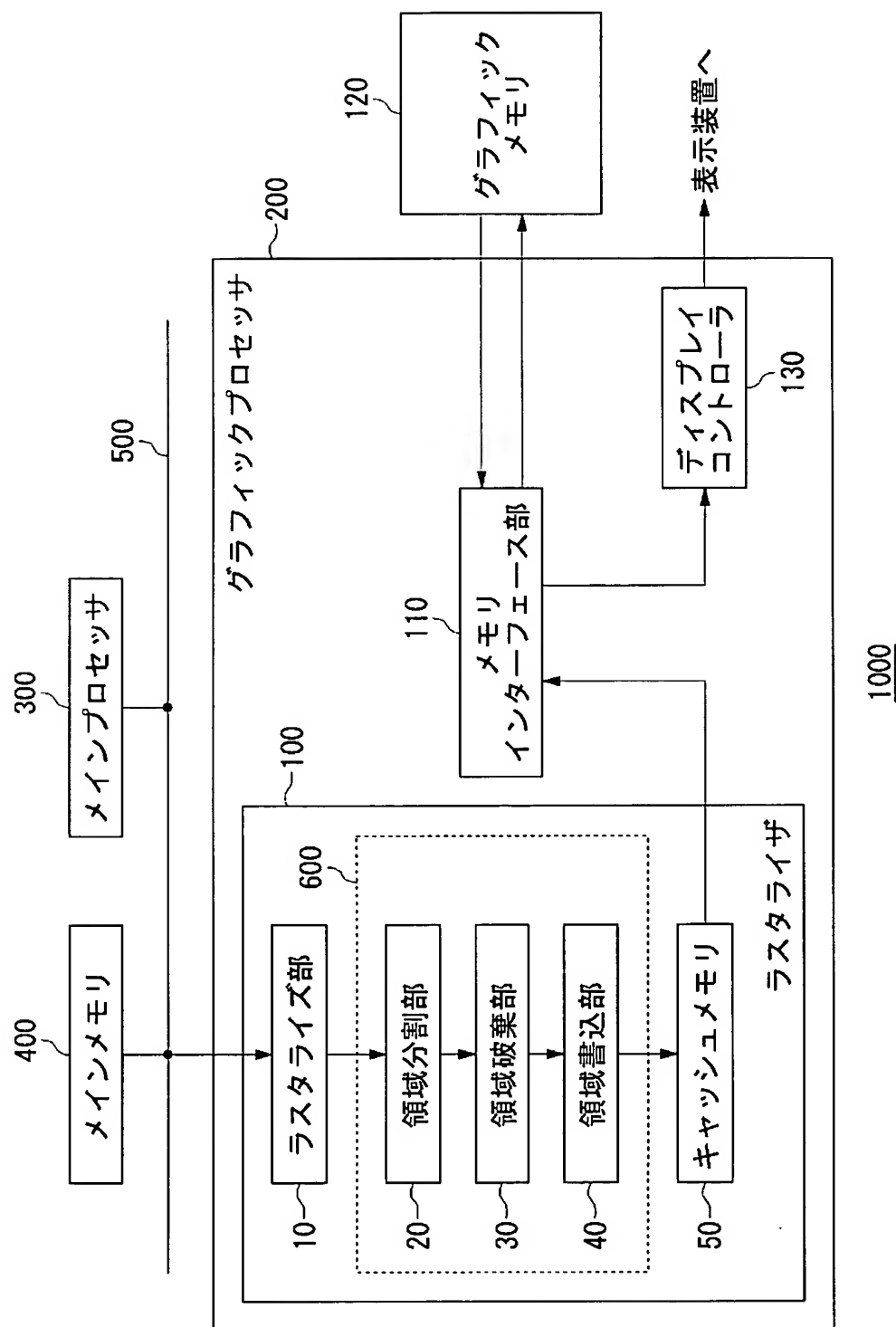
サブ領域のうち、有効なピクセルを含むサブ領域を抽出して統合領域を生成するステップと、

を実行させることを特徴とするコンピュータプログラム。

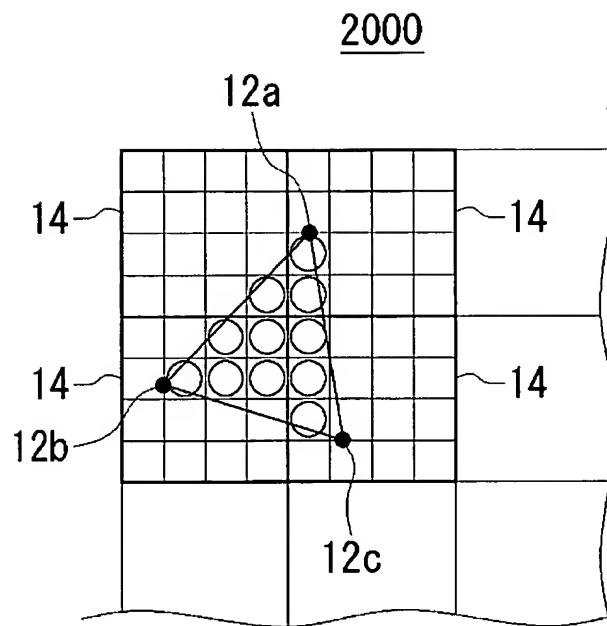
- [19] 描画対象の三次元オブジェクトのサーフェスを構成する単位図形ごとにスクリーン座標系にて順次描画処理するコンピュータプログラムであって、コンピュータに、  
単位図形をスクリーン座標上で複数の単位領域に分割して出力するステップと、  
前記出力するステップにおいて出力される単位領域のそれぞれを構成する複数のサブ領域のうち、有効なピクセルを含むサブ領域を並行的にメモリに書き込むステップと、

を実行させることを特徴とするコンピュータプログラム。

[図1]

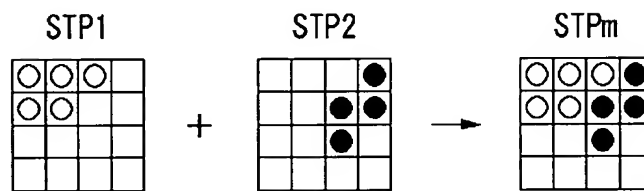


[図2]

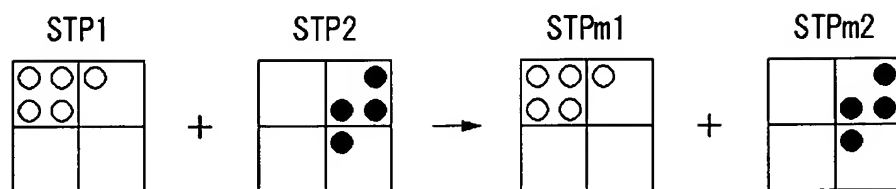


[図3]

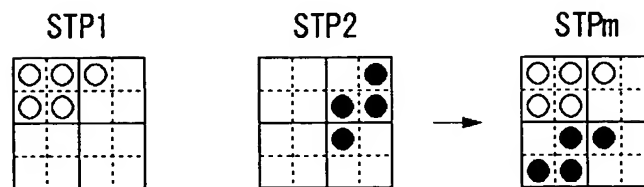
(a)



(b)

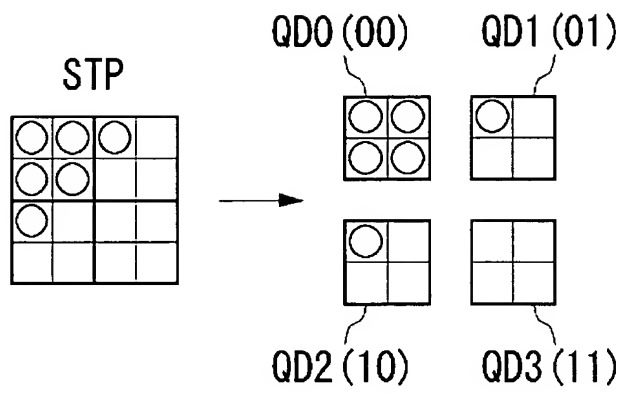


(c)

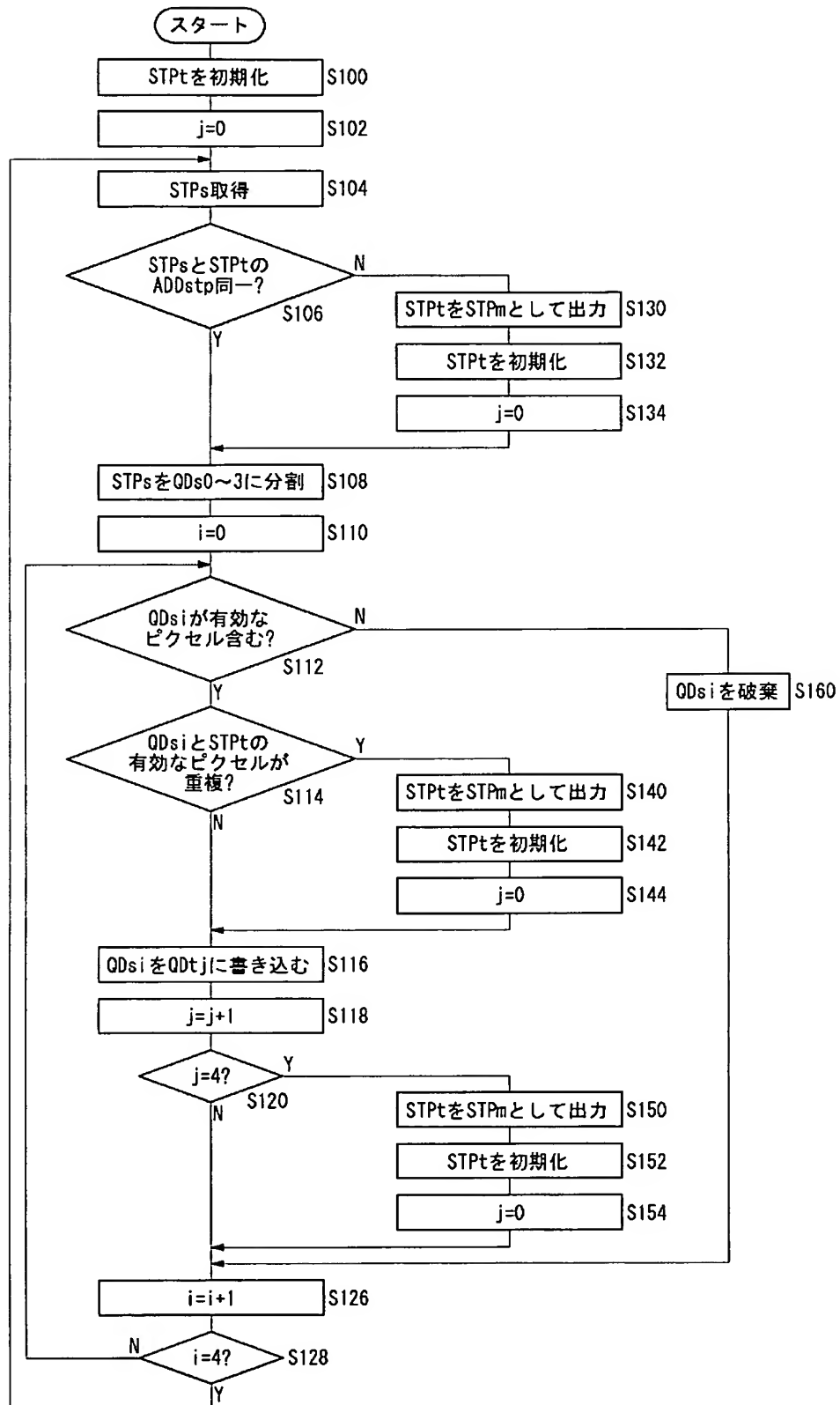




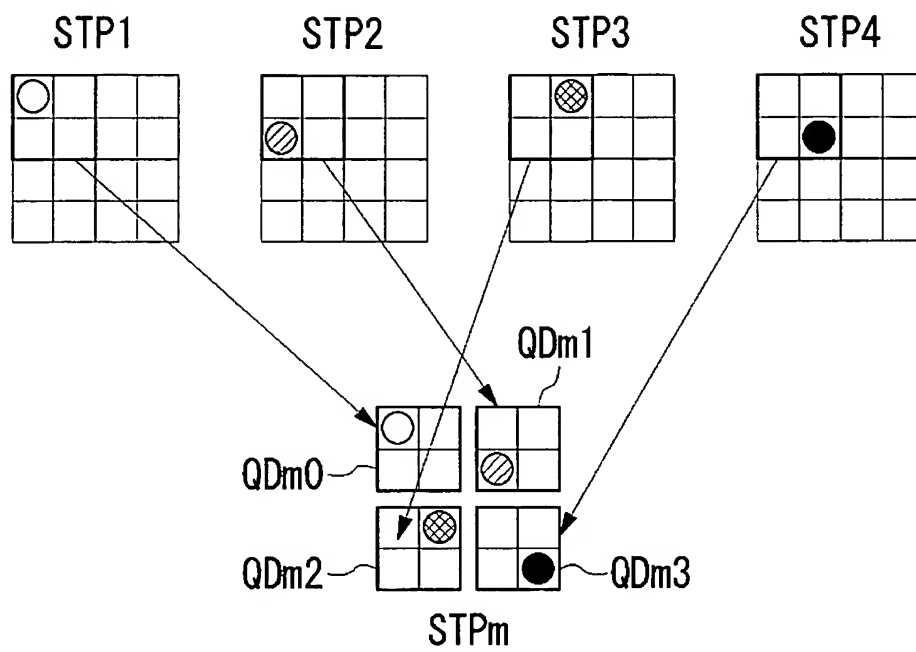
[図4]



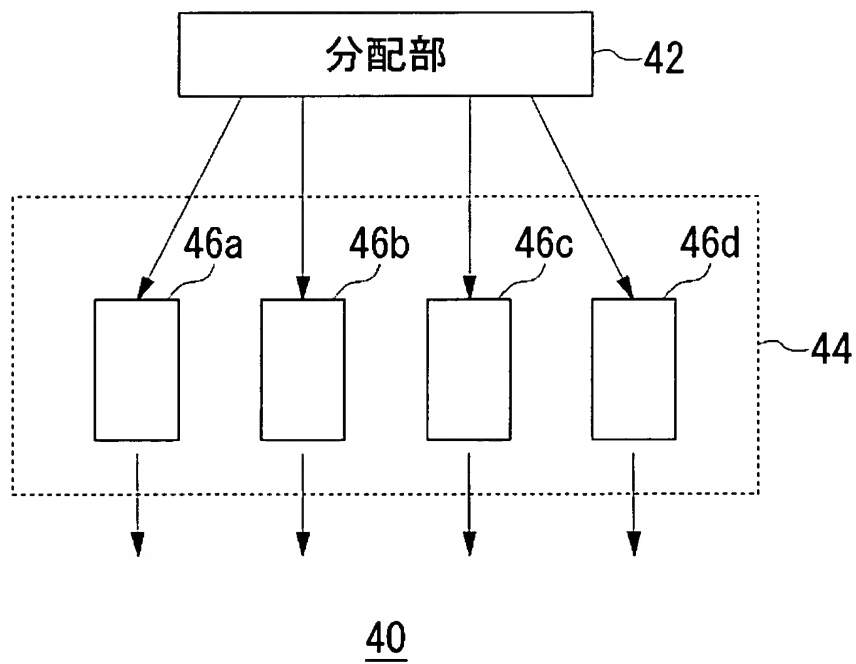
[図5]



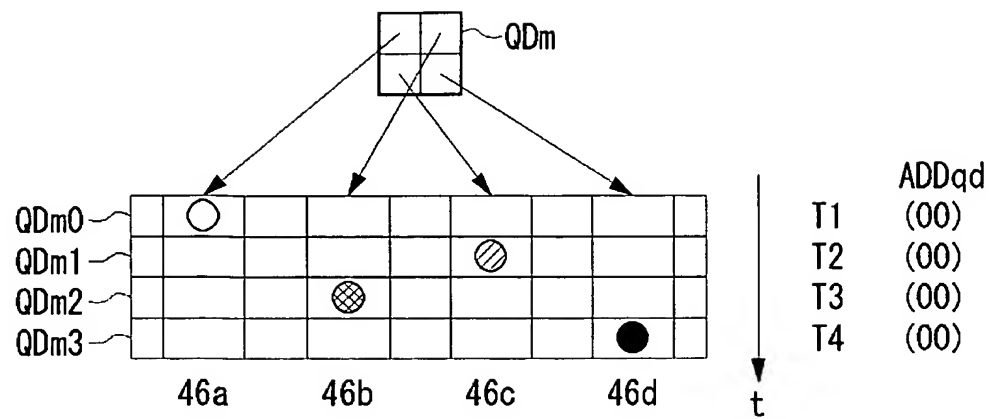
[図6]



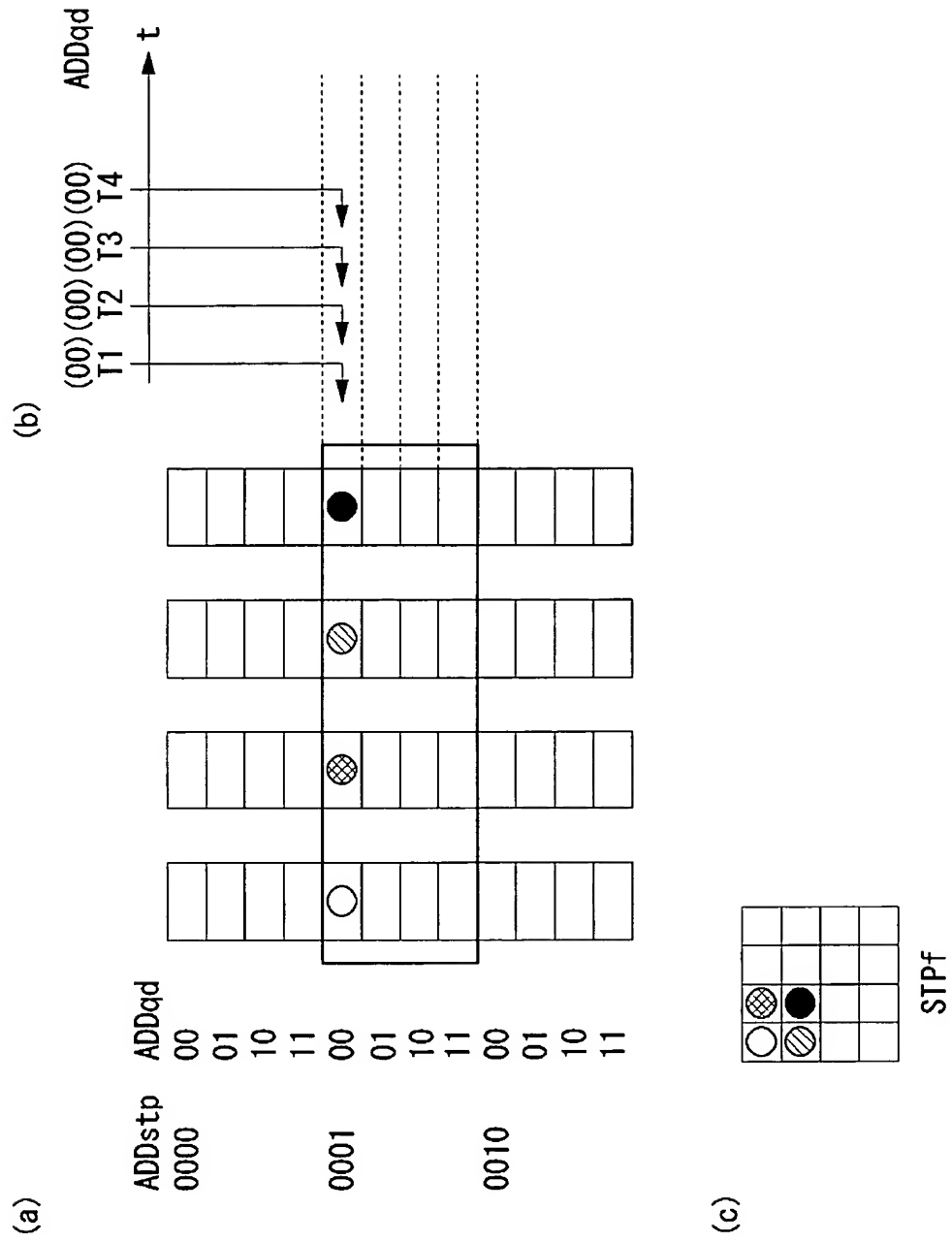
[図7]



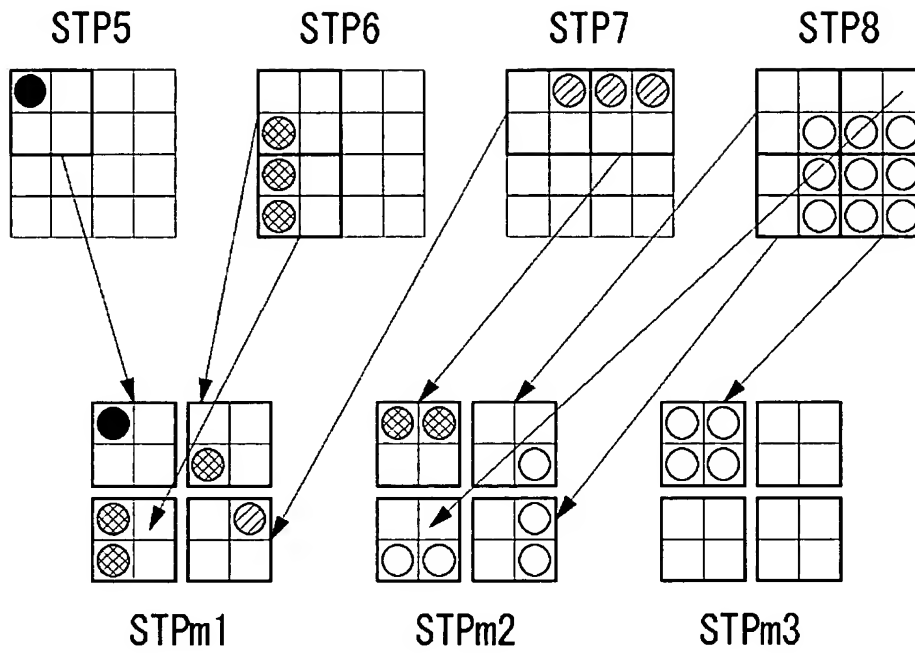
[図8]



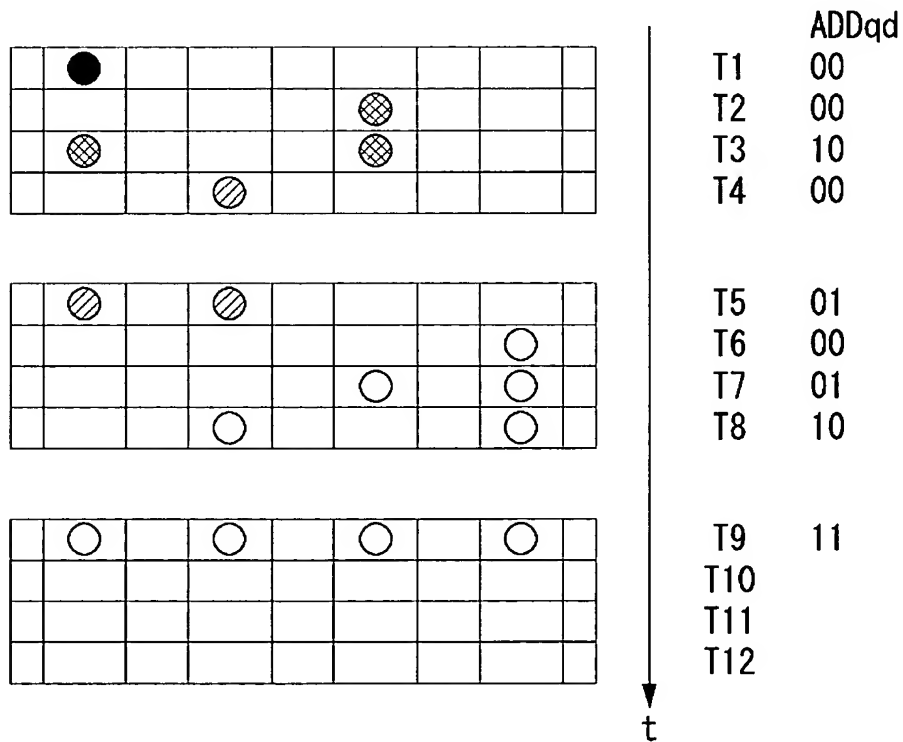
[図9]



[図10]

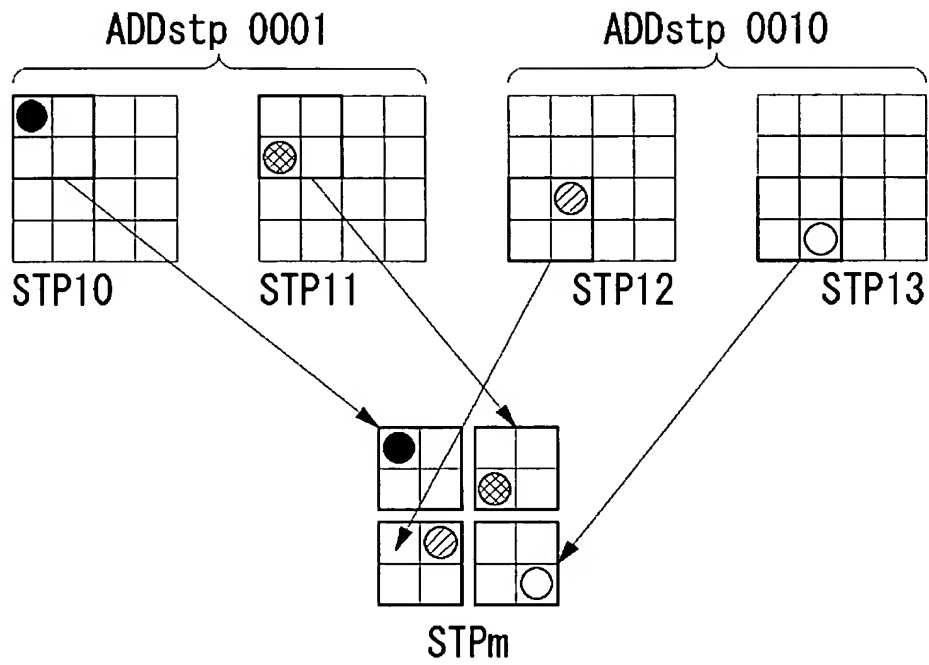


[図11]

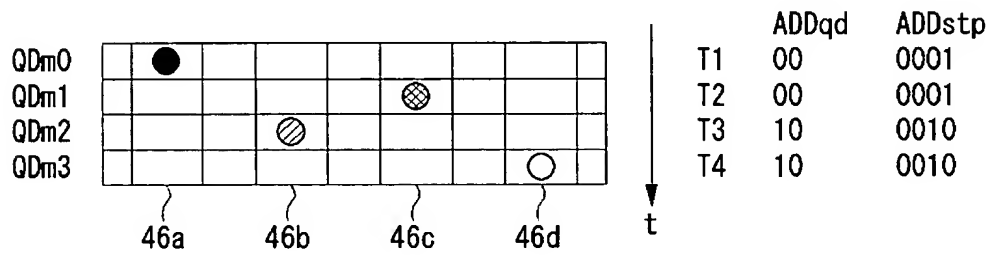




[図13]



[図14]





[図15]

